EPJ Web of Conferences **380**, 02016 (2026) https://doi.org/10.1051/epjconf/202638002016
ICISCT'25

Enhanced Private key Synchronization Mechanism Security in Passkeys System Based on Elliptic Curve Diffie-Hellman and Zero-Knowledge Proofs

Assane ILBOUDO[1,2,3,*], *Didier* BASSOLE[1,2,3,**], and *Désiré* GUEL[1,2,3,***]

[1]*dept. of Computer Science*
[2]*University Joseph KI-ZERBO*
[3]Ouagadougou, Burkina Faso

Abstract. Based on asymmetric cryptography, Passkeys Systems are a secure authentication method that can serve as an alternative to traditional authentication methods, such as usernames and passwords. In this paper, we propose a secure approach to enhance private key synchronization mechanisms in passkeys systems. Our secure service is based on Elliptic Curve Diffie-Hellman protocol and Zero-Knowledge Proofs in a peer-to-peer environment. Following a critical analysis of existing works, which highlights recurrent vulnerabilities related to authentication and confidentiality, we introduce a robust architecture using mutual identity verification, secure session key generation and encrypted passkey transfer. The security of our proposed protocol is assessed through a dual approach: an informal analysis based on potential attack modeling, and a formal validation using ProVerif and Scyther tools. The results demonstrate enhanced resistance to replay attacks, man-in-the-middle attacks, message modification, and identity impersonation, while ensuring optimized performance in terms of computational and communication costs.
keywords: Elliptic Curve Diffie-Hellman, Zero-Knowledge Proofs, Informal Analysis, Formal Analysis, Security

1 Introduction

Direct communication between devices eliminates the need for a centralized infrastructure. In this process, data is transmitted from device to device without an intermediary. However, an analysis of recent works reveals persistent security vulnerabilities, notably the absence of robust authentication mechanisms and inadequate management of confidentiality. In response to these challenges, we propose a new passkey synchronization mechanism that leverages the capabilities of Elliptic Curve Diffie-Hellman (ECDH) and Zero-Knowledge Proofs (ZKP) to establish direct, secure communications resistant to interception and impersonation attacks. This work reflects a commitment to reconciling operational simplicity with a high level of security in peer-to-peer environments.

The remainder of this article is organized as follows: Section 2 reviews related work; Section 3 describes our methodological approach; Section 4 presents an informal security

*e-mail: assaneilboudo1998@gmail.com
**e-mail: dbassole@gmail.com
***e-mail: guel.desire@gmail.com

analysis of our mechanism; Section 5 details the formal security analysis; Section 6 is dedicated to the performance evaluation of our approach. Finally, Section 7 concludes the study.

2 Related Work

Saba Fatima [1] proposes a messaging mechanism that combines Wi-Fi Direct and RSA cryptography. This mechanism establishes direct communication between devices without an intermediary. The author does not describe in detail the public key sharing process, which is essential for the security of the communicating parties. Blessing et al. [2] propose a cloud-based mechanism for synchronizing private keys. This mechanism solves the synchronization problem in passkey authentication systems. This solution raises security risks regarding key confidentiality. Khawaja et al. [3] propose a collaborative approach that combines Wi-Fi Direct and Bluetooth in an ad hoc system. This approach combines the performance advantages of Wi-Fi (high throughput and fast transmission speed) and Bluetooth (low power consumption), but lacks a security mechanism. Dang et al.[4] propose an Enhanced Wireless Direct Connectivity (EWDC) approach that combines DECT-2020 and Wi-Fi to optimize wireless communication. This approach replaces Bluetooth Low Energy—which is vulnerable within DECT-2020—for the device pairing phase. Wu et al. [5] propose a formal analysis methodology based on ProVerif to evaluate the security of Bluetooth protocols. This analysis identifies 82 security violations. Ayoub et al. [6] demonstrate the feasibility of a "screaming channels" attack on the Bluetooth Low Energy protocol. This attack allows cryptographic keys to be extracted through electromagnetic emissions. This highlights the vulnerabilities of critical hardware without requiring direct interaction with the devices. Niebla-Montero et al. [7] propose an Opportunistic Edge Computing (OEC) architecture based on Bluetooth 5 and Wize to improve RFID communications in vehicular environments. This architecture combines the performance of Bluetooth 5 (low power consumption) and Wize (long-range communication capability). Yadav et al. [8] propose an authentication protocol for the metaverse. This protocol is based on zero-knowledge proofs (ZKPs) and blockchain technology to secure communication between users and the platform. Pirmoradian et al. [9] present an ECKCI key exchange protocol based on elliptic curves (ECC). This protocol enhances the security of communications in telemedicine systems (TMIS) and the Internet of Medical Things (IoMT). Ghani et al. [10] introduce a lightweight MCDH-SKLAP key exchange protocol based on the classic Diffie-Hellman (CDH) approach. This protocol secures communications within decentralized edge computing networks (DECN). Hassan et al. [11] propose an authentication scheme for the metaverse. This scheme combines blockchain, elliptic curve cryptography (ECC), biometric hashing, and physically unclonable functions (PUFs). This solution ensures decentralization, interoperability, and privacy protection.

3 Methodological approach

An in-depth analysis of recent research on direct device-to-device communication systems. This analysis highlights a variety of approaches using technologies such as Bluetooth, Wi-Fi Direct, cloud computing, and DECT-2020. These architectures enable data exchange without an intermediary. The proposed solutions reveal vulnerabilities, including the absence of robust authentication mechanisms and public-key sharing protocols. Furthermore, the risks of identity theft and side-channel attacks are largely underestimated, particularly in hybrid architectures that combine multiple technologies. To our knowledge, none of the solutions identified successfully reconcile the fundamental security requirements of confidentiality, integrity, authentication, and resistance to identity theft.

We propose a mechanism for synchronizing access keys across devices. This mechanism is based on peer-to-peer communication, Diffie-Hellman elliptic curves (ECDH), and zero-knowledge proofs (ZKP). This process consists of four (04) steps, as shown in the figure 1:

1. **Pairing Initialization**,
2. **Device Identity Verification**,
3. **Secure Session Key Generation**,
4. **Direct and Secure Transfer of Passkeys**.

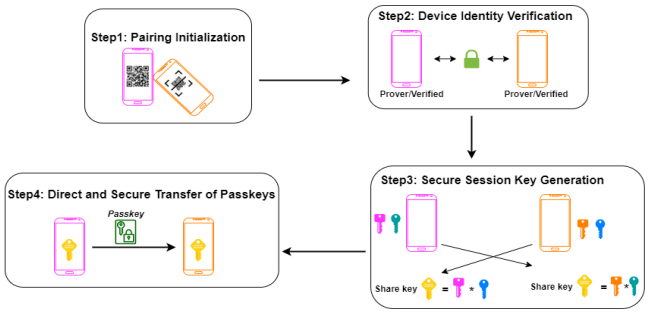

Figure 1. Steps in Our Synchronization Mechanism

The proposed architecture describes the process of synchronizing passkeys between two (02) devices, A and B. This process is shown in Figure 2.

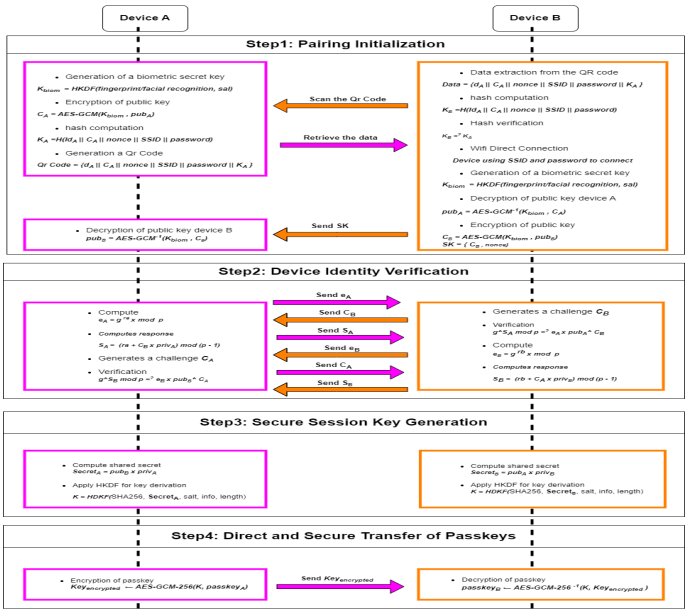

Figure 2. Description of Our Architecture

3.1 Pairing Initialization

Device A generates a secret biometric key K_{biom} derived from the `HKDF` function using a biometric factor (fingerprint or facial recognition) combined with a random salt. The key K_{biom} is used to encrypt the public key pub_A using the AES-GCM algorithm to obtain a ciphertext C_A. A digest K_A is then calculated from the concatenation of the following elements {a unique identifier, the ciphertext C_A, a random nonce, the network's SSID, and a password }. These elements are encapsulated in a QR code, which is scanned by device B. Device B extracts the data from the QR code. It recalculates a digest K_B and compares it to K_A to verify the integrity of the data. If the verification succeeds, B uses the extracted information (SSID and password) to establish a Wi-Fi Direct connection. It then generates its own biometric key K_{biom}, which it uses to decrypt C_A and retrieve A's public key. In turn, B encrypts its own public key pub_B to produce C_B, then transmits the pair $\{C_B, \text{nonce}\}$ to device A, which decrypts C_B to obtain pub_B.

3.2 Device Identity Verification

In the second step, we use the Schnorr protocol for mutual authentication between the two devices, A and B. Each mechanism generates a commitment in the form of

$$e = g^r \bmod p$$

where r is a random integer and g is a generator of a cyclic group modulo p. Device A computes a response S_A by combining its private key with a challenge received from B, while B performs a similar computation. Each party then verifies the other's commitment by checking that the following equality holds

$$g^S \equiv e \cdot \text{pub}_C \bmod p$$

This equation allows for mutual verification. Each device confirms that the other indeed holds the private key associated with its public key. No private keys are revealed.

3.3 Secure Session Key Generation

In this step, each device uses the other's public key and its own private key to derive a shared secret Secret_{AB} using the ECDH protocol. This secret is then transformed using the HKDF function with SHA-256 into a symmetric session key K, which will be used to encrypt sensitive communications between the two devices.

3.4 Direct and Secure Transfer of Passkeys

The final step consists of securely transferring the passkey. Device A encrypts the passkey using the session key K and the AES-GCM-256 algorithm, producing an encrypted message referred to as $\text{Key}_{\text{encrypted}}$. This message is sent to device B, which uses the same session key to decrypt it and retrieve the passkey initially held by A.

4 Informal security analysis

The informal method is a security analysis approach based on intuitive reasoning, the analyst's creativity, and the use of mathematical concepts. Its goal is to identify potential vulnerabilities and provide arguments supporting security, without relying on a rigorous and systematic proof.

4.1 Resistance to Replay Attack

The proposed synchronization approach relies on the integration of unique nonces dynamically generated for each session, used both in the construction of QR codes and during mutual authentication. These nonces ensure that each exchanged message is strictly bound to a specific session, rendering any attempt to reuse previously intercepted messages ineffective. The systematic verification of nonces allows for the automatic rejection of any message that does not match the expectations of the current session. This mechanism therefore guarantees the integrity and authenticity of the exchanged messages.

4.2 Resistance to Man-in-the-middle Attack

Our synchronization protocol relies on the exchange of pre-authenticated public keys, combined with mutual authentication based on Schnorr-like zero-knowledge proofs. Additionally, communications are encrypted using keys derived from biometric data, preventing an adversary from injecting or intercepting valid messages without detection.

4.3 Resistance to Modification Attack

The critical components of our synchronization mechanism are protected by AES-GCM encryption and robust challenge-response mechanisms. Any attempt to modify the message immediately invalidates it. This synchronization process secures the data.

4.4 Resistance to Passive Insider Secret

Partial access to a device's internal data is not sufficient. The attacker cannot derive the session key or extract the passkey. The attacker would need to possess the biometric key, the private key of the paired device, and the session's ephemeral nonces all at the same time. Sensitive data is generated from volatile information. It is never stored in plaintext.

4.5 Resistance Identity impersonation

Mutual authentication relies on the devices' possession of cryptographic secrets. Each challenge requires valid proof of this possession. An attacker who does not possess these secrets cannot respond correctly. Any attempt at impersonation therefore fails. Only a legitimate entity can initiate or validate a synchronization session.

5 Formal security analysis

The formal method can be applied manually or automatically. It relies on security analysis tools and mathematical logic frameworks. The manual approach uses logical models such as the random oracle or BAN logic. The automated approach relies on tools such as AVISPA, Scyther, or ProVerif. Here, we use Scyther and ProVerif. They validate the results obtained through informal analysis.

5.1 Robustness Evaluation Using ProVerif Tool

We evaluated the security of our synchronization approach using the ProVerif tool. The goal is to measure its robustness against malicious attacks. This section presents a formal analysis of the protocol based on an automated verification. Three risks are examined, including data leakage, message tampering, and session key compromise. The analysis is conducted within the Dolev-Yao model framework, adopted by the ProVerif tool. This model assumes an adversary with full control over the communication channel: they can intercept, alter, delete, or inject messages arbitrarily. However, the cryptographic primitives are considered ideal, meaning they are unbreakable without possession of the appropriate keys. The simulation conducted with ProVerif allowed us to model the various sensitive components of our protocol, such as public/private key pairs, session keys, and passkeys, verifying that they satisfy the fundamental properties of confidentiality, integrity, and authenticity. During the execution of the model, two outcomes are possible: if no attack can be formally demonstrated, ProVerif displays the message **RESULT[query] is true**; on the other hand, if a security breach is detected, the tool returns **RESULT[query] is false** illustrated in the algorithm 1.

Algorithm 1 ProVerif Query Evaluation

```
Run ProVerif with the specified security query
if no attack is found then
    Display RESULT[query] is true
else
    Display RESULT[query] is false
end if
```

In the course of our analysis, ProVerif consistently returned the value RESULT[query] is true, indicating that no vulnerability was identified regarding the specified security properties. The detailed results are shown in figure 3. Consequently, our synchronization mechanism meets the security requirements within the considered attack model and ensures the confidentiality, integrity, and authenticity of the exchanged data. These results thus confirm the validity of the previously established conclusions through non-formal analysis, providing rigorous proof through automated means.

```
-- Query not attacker(s[]) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query not attacker(s[])
RESULT not attacker(s[]) is true.
-- Query event(termDeviceA(x_2,y_1)) ==> event(acceptsDeviceB(x_2,y_1)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query event(termDeviceA(x_2,y_1)) ==> event(acceptsDeviceB(x_2,y_1))
RESULT event(termDeviceA(x_2,y_1)) ==> event(acceptsDeviceB(x_2,y_1)) is true.
-- Query inj-event(termDeviceB(x_2)) ==> inj-event(acceptsDeviceA(x_2)) in process 1.
Translating the process into Horn clauses...
Completing...
Starting query inj-event(termDeviceB(x_2)) ==> inj-event(acceptsDeviceA(x_2))
RESULT inj-event(termDeviceB(x_2)) ==> inj-event(acceptsDeviceA(x_2)) is true.

--------------------------------------------------------------
Verification summary:

Query not attacker(s[]) is true.

Query event(termDeviceA(x_2,y_1)) ==> event(acceptsDeviceB(x_2,y_1)) is true.

Query inj-event(termDeviceB(x_2)) ==> inj-event(acceptsDeviceA(x_2)) is true.
```

Figure 3. Summary of ProVerif simulation results for the proposed synchronization approach

5.2 Robustness Evaluation Using Scyther Tool

In this section, we use the Scyther tool to evaluate the robustness of our synchronization approach. Scyther is based on the formal Dolev-Yao model, in which the attacker is assumed to have full control over the communication channel but is unable to break perfect cryptographic primitives. The analysis assumes an unlimited number of protocol sessions, allowing us to assess the behavior of the approach in a highly concurrent environment. Scyther enables the automated verification of several fundamental security properties, including confidentiality, integrity, authenticity, and availability. These properties are expressed through various security claims, which can be either predefined or user-specified. Among the most commonly used claims in Scyther, we distinguish:

- **Alive**: this claim ensures that all protocol participants are active and available to engage in communication. It thus verifies the availability of the protocol;
- **Nisynch** (Non-injective Synchronization): this claim verifies that messages sent are indeed received by the intended party, without modification or reordering;
- **Confidentiality**: This assertion ensures that certain sensitive data is not accessible to unauthorized parties;
- **Weakagree**: This assertion confirms that the participants have reached an agreement on specific exchanged values. It helps ensure the authenticity of the shared information.

We conducted an analysis of our synchronization mechanism using Scyther, incorporating these various security assertions. The simulation results shown in Figure 4 indicate that Scyther did not detect any violations of the specified security properties. This demonstrates that our synchronization mechanism is robust against attacks in the Dolev-Yao model. These results confirm the validity of the conclusions previously established by informal analysis by providing a rigorous, automated proof.

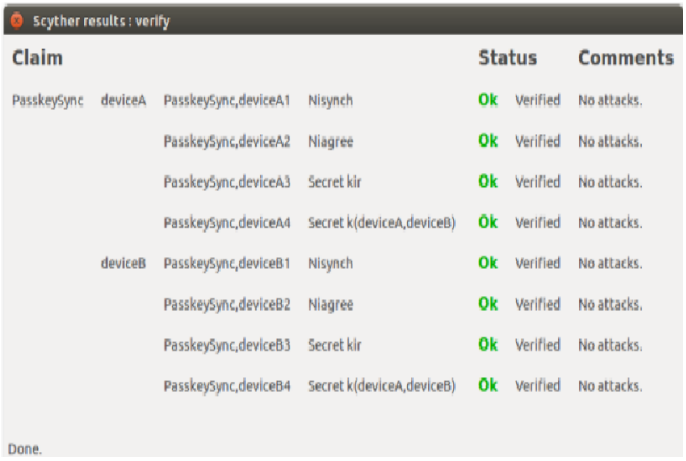

Figure 4. Results of the Scyther assertions for our proposed mechanism

5.3 Comparaison

This section presents a comparative analysis. It compares several approaches from the literature ([2], [3], [8]–[11]) with our own solution. Five (05) fundamental security criteria form

the basis of the comparison, including replay attacks, man-in-the-middle attacks, modification attacks, passive disclosure of secrets by an insider, and identity theft. The results are shown in Table 1.

Table 1. Comparison of Security Features Across Different Approaches

Feature	[2]	[3]	[8]	[9]	[10]	[11]	Ours
Replay Attack	–:	X	V	V	V	V	V
Man-in-the-middle Attack	X	X	V	V	V	V	V
Modification Attack	V	X	V	V	V	V	V
Passive Insider Secret	V	X	–:	V	–:	V	V
Identity Impersonation	X	X	V	–:	V	V	V

The comparison reveals significant weaknesses in several existing approaches. Two shortcomings in particular stand out: the lack of protection against replay attacks and against passive internal threats. Our solution covers all identified attack vectors. It holds up where previous work fails or explicitly omits certain threats. This evaluation confirms the relevance of our mechanism in environments with high security requirements. The symbols in Table 1 are interpreted as follows: **V** indicates resistance to the threat; **X** indicates a vulnerability; **–** indicates that the aspect has not been addressed.

6 Performance Evaluation

This section evaluates the performance of our synchronization mechanism. We compare it to existing solutions [8], [9], [10], and [11]. The comparative analysis focuses on computational costs, communication and storage overheads, as well as energy consumption. All experiments were carried out on a hardware platform equipped with an Intel Core i7 processor running at 2.59 GHz, 32 GB of RAM, and operating under Windows 11 Pro, version 23H2. The execution time of the primitive operations was measured according to the methodological protocol described in 2. All measurements were performed using the MIRACL cryptographic library, ensuring a precise and reproducible evaluation of performance.

6.1 Computation Cost

The objective of this section is to evaluate the computational cost of our synchronization approach and to compare it with that of existing methods in Figure 5. The experiment was repeated 100 times to ensure reliable measurements, and the average execution time was calculated, as presented in Table 2. The terms used in Table 2, namely TQC, TH, TAGE, TAGD, TPM, TP, TV, and THKDF, respectively correspond to the average execution times for the following operations: QR code generation (TQC), hash function computation (SHA-256) (TH), AES-GCM encryption (TAGE), AES-GCM decryption (TAGD), elliptic curve multiplication for ECDH (TPM), Zero-Knowledge Proof generation (prover) (TP), Zero-Knowledge Proof verification (verifier) (TV), and session key derivation via HKDF (THKDF).

Table 2. Estimated Computational Cost for a Synchronization Session

Operation	Estimated Time (ms)	Occurrences per Session	Total Time (ms)
TQC	5	1	5
TH	0.034	4	0.136
TAGE	0.030	2	0.060
TAGD	0.030	2	0.060
TPM	0.485	2	0.970
TP	0.6	1	0.6
TV	0.6	1	0.6
THKDF	0.050	1	0.050

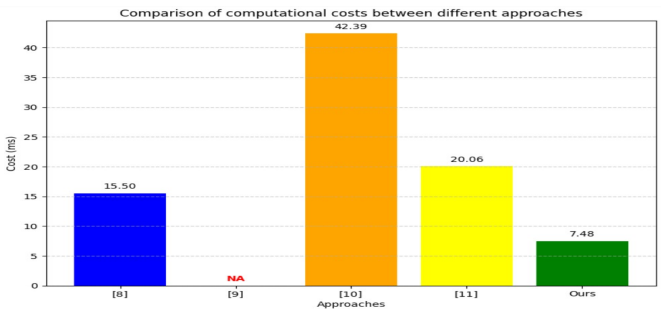

Figure 5. Comparison of computational costs between different approaches

6.2 Communication Cost

In this section, we present the communication cost of our synchronization approach, evaluated in comparison with existing methods, as illustrated in Figure 6. The bit sizes of the different elements are as follows: 128 bits for the nonce, 512 bits for the ZKP (Schnorr proof), 160 bits for the hash digest, and 256 bits for the compressed ECDH key. Four messages are exchanged between device A and device B:

- **M1** = {QR Code},
- **M2** = {Commitment (CB), Nonce},
- **M3** = {ZKP},
- **M4** = {Encrypted Passkey}.

The communication cost of our approach is detailed in Table 3. Thus, a total of **1670 bits** is required to synchronize devices A and B and to ensure the secure transfer of the passkey from device A to device B.

Table 3. Communication cost per message.

Message	Size (bits)
M1 (compressed data)	390 bits
M2	256 + 128 = 384 bits
M3	512 bits
M4	256 + 128 = 384 bits

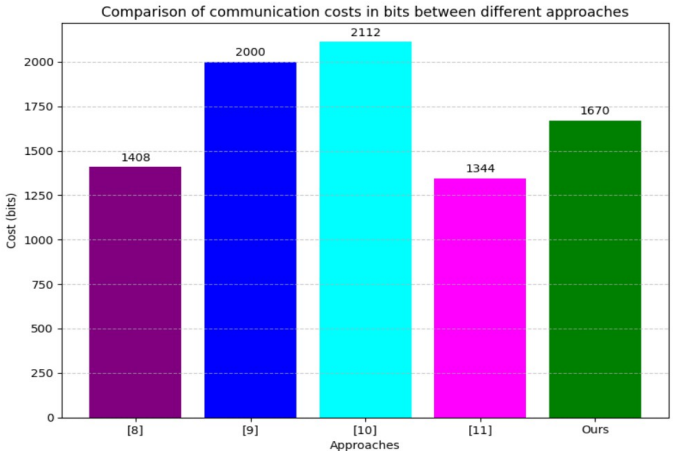

Figure 6. Comparison of communication costs in bits between different approaches

6.3 Energy Consumption

In this section, we present the estimated energy consumption required by our synchronization mechanism. The energy consumption is evaluated by considering both the number of cryptographic operations performed and the amount of data transmitted. The energy consumed by these cryptographic operations and the data transmission is estimated through simulation using a StrongARM processor running at 133 MHz and executing various representative tasks. The energy costs associated with each operation are summarized in Table 4.

Table 4. Energy Consumption for Different Operations

Element	Quantity	Energy per Unit	Total Energy
Transmission Bits	1670 bits	0.00066 mJ/bit	1.10 mJ
SHA-256	4	0.000108 mJ	0.000432 mJ
ECC Multi-plication (ECDH)	4	8.80 mJ	35.20 mJ
AES-GCM	4	0.00217 mJ	0.00868 mJ
HKDF	1	~0.002 mJ (est.)	0.002 mJ
ZKP	2	12.00 mJ	24.00 mJ

Based on these values, the total energy consumption of our synchronization approach is calculated as follows:

$$\begin{aligned} EC &= (1670 \times 0.00066) + (4 \times 0.000108) + (4 \times 8.80) \\ &\quad + (4 \times 0.00217) + (1 \times 0.00200) + (2 \times 12.00) \\ &\approx 60.31 \text{ mJ} \end{aligned} \tag{1}$$

This value is also reported in Table 4. Finally, we compare the energy consumption of our model with that of Yadav et al. [8]; the results of this comparison are shown in Figure 7.

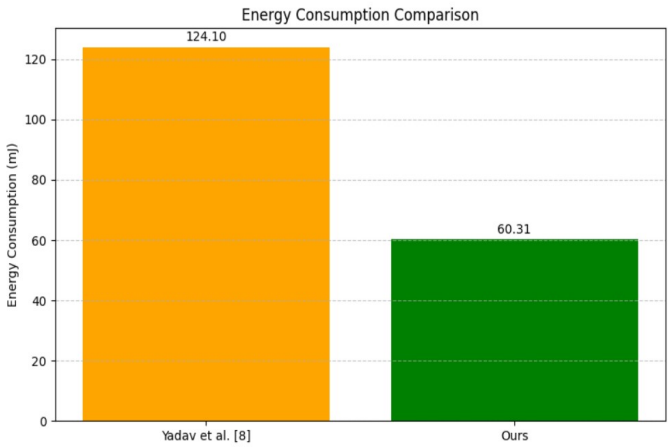

Figure 7. Energy Consumption Comparison between different approaches

6.4 Discussion

The comparative evaluation of computational costs reveals that the proposed approach significantly outperforms existing methods in both efficiency and lightweight performance.Table 5 presents the results. Our approach achieves an average cost of 7.476 ms. This is the lowest

value in the set. Comparisons confirm this. Yadav et al. [8] require 15.5 ms. Hassan et al. [11] achieve 20.057 ms. Ghani et al. [10] reach a peak of 42.386 ms. Pirmoradian et al. [9] do not provide any computational cost measurements, which excludes their solution from the quantitative analysis. Two factors account for this reduction. The choice of cryptographic primitives ECDH for key exchange, HKDF for derivation, and AES-GCM for encryption. Each critical operation executes in minimal time. An optimized ZKP protocol based on Schnorr. It reduces the computational load without sacrificing party authentication. These results point to a field of application. Our mechanism is well-suited for resource-constrained environments such as mobile devices and the Internet of Things (IoT). Performance and energy efficiency are critical in these settings. Reduced computational cost also lowers session establishment latency and optimizes energy consumption—two decisive advantages for decentralized applications. Our proposed mechanism stands out in two ways. Its robustness is formally validated in the Dolev-Yao model. Its computational load and energy consumption remain low. It thus paves the way for secure applications in mobile and heterogeneous environments.

7 Conclusion

Our passkey synchronization mechanism combines ECDH, zero-knowledge proofs (ZKPs), and authenticated encryption mechanisms. It secures data exchanges between devices. Informal analysis, supplemented by formal validation using ProVerif and Scyther, confirms its robustness. Computational cost and communication overhead remain low. This efficiency paves the way for practical adoption, from mobile communications to resource-constrained environments.

References

[1] Fatima, S. (2024). *Secure Relay Chat System based on Gossip Protocol using Flutter*. International Journal of Interpreting Enigma Engineers (IJIEE), 1(4), 7-13. https://ejournal.svgacademy.org/index.php/ijiee/article/view/91

[2] Blessing, J., Hugenroth, D., Anderson, R. J., & Beresford, A. R. (2024). *SoK: Web Authentication in the Age of End-to-End Encryption*. arXiv preprint arXiv:2406.18226. https://arxiv.org/pdf/2406.18226

[3] Khawaja, I. A., Abid, K., Farooq, U., Malik, Z., & Abid, A. (2024). *Empowering Collaborative Application Development: A Robust Framework for Ad-Hoc Distributed Systems*. IEEE Access.

[4] Dang, Y., Quang, N., Roni, F., Veli-Matti, R., Kalle, R., & Jäntti, R. (2025). *EWDC: Integrating DECT-2020 with Wi-Fi for Enhanced Wireless Direct Connectivity. Authorea Preprints*. https://d197for5662m48.cloudfront.net/documents/publicationstatus/241045/preprint_pdf/7c99fa154dfab61eb4b092f58d1f2ea9.pdf

[5] Wu, J., Wu, R., Xu, D., Tian, D. J., & Bianchi, A. (2022, May). *Formal model-driven discovery of bluetooth protocol design vulnerabilities*. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 2285-2303). IEEE. https://www.cs.purdue.edu/homes/dxu/pubs/SP22_BT_formal.pdf

[6] Ayoub, P., Cayre, R., Francillon, A., & Maurice, C. (2024, December). *BlueScream: Screaming Channels on Bluetooth Low Energy*. In 40th Annual Computer Security Applications Conference (ACSAC'24). https://hal.science/hal-04725668v2/file/bluescream.pdf

[7] Niebla-Montero, A., Froiz-Míguez, I., Fraga-Lamas, P., & Fernández-Caramés, T. M. (2024, June). *Practical Evaluation of Wize and Bluetooth 5 Assisted RFID for an Opportunistic Vehicular Scenario*. In 2024 IEEE International Conference on RFID (RFID) (pp. 101-106). IEEE. https://arxiv.org/pdf/2410.23366

[8] Yadav, A. K., Braeken, A., Ylianttila, M., & Liyanage, M. (2023, June). *A blockchain-based authentication protocol for metaverse environments using a zero knowledge proof*. In 2023 IEEE international conference on metaverse computing, networking and applications (MetaCom) (pp. 242-249). IEEE. https://oulurepo.oulu.fi/bitstream/handle/10024/44657/nbnfi-fe20231031142009.pdf?sequence=1

[9] Pirmoradian, F., Dakhilalian, S. M., & Safkhani, M. (2024). *ECKCI: An ECC-Based Authenticated Key Agreement Scheme Resistant to Key Compromise Impersonation Attack for TMIS*. ISeCure, 16(2).

[10] Ghani, A., Jan, S. U., Chaudhry, S. A., Ahmad, R., & Kim, D. H. (2024). *MCDH-SLKAP: Modified Computational Diffie-Hellman based Secure and Lightweight Key Agreement Protocol for Decentralized Edge Computing Networks*. IEEE Access. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10680027

[11] ul Hassan, M., Abbas, Y., Iqbal, W., Chehri, A., & Iqbal, J. (2025). *PRIDA-ME: A Privacy-Preserving, Interoperable and Decentralized Authentication Scheme for Metaverse Environment*. IEEE Open Journal of the Communications Society. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10819498